Product Clustering for Online Crawlers

Anonymous Author(s)
Affiliation
Address
email

Abstract

One of the biggest challenges of the E-commerce solutions is dealing with the huge amount of products available online. Understanding the available data and finding similarity between the products is crucial when dealing with scalability issues. In this paper, we are proposing a modified agglomerative information bottleneck method to cluster online products and recognize the same products on different merchants websites. Recognizing similar products on different merchants sites makes a huge difference in the way the product is presented to the costumers on the aggregator websites such as Kijiji, or Wishpond.

1 Introduction

E-commerce solution providers who integrate online products from different websites, encounter the problem of clustering similar products from different sources. Product clustering is different than clustering of web-pages or online auctions or data streams. Product clustering deals with large amount of data; moreover, the data is not in the form of Eaucleadian distance vectors which is the input of many clustering algorithms. A simple example of the above problem is displayed in table1. One might think having different version of a product is not a problem but then when dealing with large number of products, server cost for crawling and tracking is going to be out of hand.

Product clustering can be seen as a text clustering problem. Most of the clustering algorithms have two approaches. The first approach is based on the pairwise distance of points. The second approach is based on the distortion measure between a single point and center of a class. The first approach is called pairwise clustering while the second approach is called vector quantization. The cost function for the pairwise clustering or vector quantization method tries to minimize the intra-class distortion or maximize the intra-class connectivity. The problem with pairwise clustering and the vector quantization methods is that the distance or distortion measurements are subjective and cannot be a good choice when dealing with large amount of data. To solve the problem of distance or distortion measurement, Tishby *et. al* proposed a method that given the joint distribution of two random variables *p(x,y)*, compute the compressed form of variable *X* in a way that keeps the important features on relevance of variable *X* and *Y*. Tishby method is called information bottleneck method.

1.1 Information Bottleneck Method

When dealing with information theory, the notion of *meaningful* is not well defined due to the Shannon theory definition. Shannon theory only cares about data transmission and the content of transmitted data is not important. The common belief of statistical data and communication theory disregards the meaning or relevance of data. Tibshy *et. al*[1] proved that communication theory specially the theory of lossy source compression is a decent foundation in providing relevant information. Relevant information plays a key role in clustering.

Table 1: Same product with different title is integrated from 4 different sources

Product title	Source
Sony BRAVIA KDL40BX420 40-Inch 1080p LCD HDTV, Black	Amazon
Sony Bravia 40” Class LCD 1080p 60Hz HDTV, KDL-40BX420	Walmart
Sony 40” Class 1080p 60hz LCD HDTV - Black (KDL40BX420)	Target
Sony - BRAVIA / 40” Class / 1080p / 60Hz / LCD HDTV	Best Buy

Extracting relevant information requires knowing the important features. The lossy source compression theory is one way of approaching the problem. Lossy source compression uses the distortion function to get the important features. Distortion function is a trade off between the distortion caused by the compression and the rate of compression or size of the compressed signal. The problem with using the distortion function is that we have to first find the distortion distribution. To find the distortion distribution, we have to find the important features of the signal. The problem of finding the important feature of a signal is a subjective selection.

To avoid the selection of bad features, Tishby *et. al* suggested to use extra information in the data as a lead to choose the right features that keeps the relevancy of information. In information bottleneck method, the data set, or variable *X*, is divided into partitions and each partition is mapped with a codeword $\tilde{x}$. The codeword is the compressed form of *X*. The stochastic mapping of *X* and $\tilde{x}$ is based on $p(\tilde{x}|X)$ where $p(\tilde{x}|X)$ is the soft partitioning of *X*. The probability of the codeword is:

$$p(\tilde{x}) = \sum p(x)p(\tilde{x}|x) \tag{1}$$

The minimum number of bits that can be used to represent a variable so that the original variable can be uniquely identified is a good measure for the quality of quantization in information bottleneck method. This minimum amount is equal to the mutual information. Mutual information is defined as follows:

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2}$$

In information bottleneck algorithm, *X* is divided into sections based on $p(\tilde{x}|X)$ mapping; therefore, the mutual information between *X* and $\tilde{x}$ can be written as follows using Bayes rule.

$$I(X,Y) = \sum_{\tilde{x} \in \tilde{X}} \sum_{x \in X} p(x,\tilde{x}) \log \frac{p(\tilde{x}|x)}{p(x)} \tag{3}$$

Looking at the above equation, one can find the resemblance between the mutual information formula and the KullbackLeibler divergence(KL). KL divergance is a measure of difference between two distributions *Q* and *p*. KL measures the extra number of bits which is required to code samples from *q* based on *Q*. Typically, *p* is the true distribution and *Q* is and estimation of *p*. KL divergence is defined as follows:

$$D_{KL} = \sum_{i} p(i) \log \frac{p(i)}{Q(i)} \tag{4}$$

To formulate the problem of finding the best quantization of *X*, Tishby[2] suggested a positive Lagrange multiplier β as the cost function:

$$Ł[p(\tilde{x}|x)] = I(X;\tilde{X}) - \beta I(\tilde{X};Y) \tag{5}$$

It is easy to understand why a Lagrange multiplier can be a good formulation for the problem. In information bottleneck theory we are interested in finding a compressed form of *X* such that the prediction of *Y* from *X* through $\tilde{X}$ will be as pricise as direct extraction of *Y* from *X*.

Solving the above Lagrangian cost function results into the following solution for $p(\tilde{x}|x)$ and $p(y|\tilde{x})$:

$$\begin{cases} p(\tilde{x}|x) = \frac{p(\tilde{x})}{Z(\beta,x)} \exp\left(-\beta D_{KL}[p(y|x)||p(y|\tilde{x})]\right) \\ p(y|\tilde{x}) = \sum_x p(y|x)p(\tilde{x}|x)\frac{p(x)}{p(\tilde{x})} \\ p(\tilde{x}) = \sum_x p(\tilde{x}|x)p(x) \end{cases} \tag{6}$$

Where $Z(\beta, x)$ is the normalization factor. In information bottleneck method, we are interested in relevance of pair of variables. We are looking for important features of a variables that can give us information about predicting other variables. This method looks for the minimum length of *X* that holds the maximum information about *Y*. In each step, variable *X* is divided into sections. During the next iteration, sections that are sharing the same information more than others are merged together until we have the desired amount of sections or clusters.

1.2 Agglomerative Clustering Algorithm

In agglomerative clustering algorithm, each data point is located in a separate cluster. The distance matrix of data set which is based on the pairwise distance is evaluated. Based on the distance matrix, clusters with the smallest distance are merged together and create a new class. In each step a pair of classes are merged together and the total number of classes decrease. The iteration continues until one class remains. In case, we are interested in having *n* clusters, we can stop the iterations when the number of sections is n.

1.3 Agglomerative Information Bottleneck(AIB)

Combining Agglomerative clustering and information bottleneck method, another method of clustering is created called agglomerative information bottleneck algorithm. In equation 6 , Tishby proposed the use of hard clustering instead of soft clustering:

$$\begin{cases} p(\tilde{x}|x) = \begin{cases} 1 & if x \in X \\ 0 & otherwise \end{cases} \\ p(y|\tilde{x}) = \sum_x p(y|x)p(\tilde{x}|x)\frac{p(x)}{p(\tilde{x})} \\ p(\tilde{x}) = \sum_x p(\tilde{x}|x)p(x) \end{cases} \tag{7}$$

Using the above probability distribution, it is easy to calculate the mutual information based on equation 3; therefore, it is possible to extract the partitions of *X*. Using agglomerative clustering algorithm, we find clusters of word which have the highest mutual information on the documents. In each step, we merge two clusters into a new cluster. We choose the clusters based on the minimum loss of mutual information in each step. The new partition, $\tilde{\star x}$, when merging $\tilde{x_i}$ and $\tilde{x_j}$ is define as :

$$\begin{cases} p(\tilde{\star x}|x) = \begin{cases} 1 & if x \in \tilde{x_i} or x \in \tilde{x_j} \\ 0 & otherwise \end{cases} \\ p(y|\tilde{\star x}) = \frac{p(\tilde{x_i})}{p(\tilde{\star x})}p(y|\tilde{x_i}) + \frac{p(\tilde{x_j})}{p(\tilde{\star x})}p(y|\tilde{x_j}) \\ p(\tilde{x}) = p(\tilde{x_i}) + p(\tilde{x_j}) \end{cases} \tag{8}$$

Loss of mutual information can be calculated as:

$$\delta I(\tilde{x_i}, \tilde{x_j}) = I(\tilde{X}_{before}; Y) - I(\tilde{X}_{after}; Y) \tag{9}$$

After some algebric manipulation, equation 9 can be re-written as:

$$D_{JS}[p_i, p_j] = \pi_i D_{KL}[p_i||\bar{p}] + \pi_j D_{KL}[p_j||\bar{p}] \tag{10}$$

In the case of agglomerative information bottleneck:

$$\begin{cases} p_i, p_j = p(y|\tilde{x_i}), p(y|\tilde{x_j}) \\ \pi_i, \pi_j = \frac{p(\tilde{x_i})}{p(\tilde{\star x})}, \frac{p(\tilde{x_j})}{p(\tilde{\star x})} \\ \tilde{p} = \pi_i p(y|\tilde{x_i}) + \pi_j p(y|\tilde{x_j}) \end{cases} \tag{11}$$

2 Modified AIB

Aggloremative information bottleneck algorithm is a great approach in dealing with documents with sufficient amount of text; therefore, we propose the modified AIB to enhance the performance of AIB on data sets with less amount of text. In modified AIB method, when finding the minimum cost in the distance matrix, instead of finding only one minimum cost, we find a set of *k* minimum cost values. Lets assume that there is a set of extra information on each of the elements in the data set,

```
Input : joint probability p(x,y)
output : partition of X into m cluster
Inititialize:
    $\tilde{X} = X$
    $d_{i,j} = (p(\tilde{x_i}) + p(\tilde{x_j}))D_{JS}[p(y|\tilde{x_i}), p(y|\tilde{x_j})]$
Loop:
    For all the partitions in X
            find the smallest $d_{i,j}$
            merge $\tilde{x_i}, \tilde{x_j}$ into $\star\tilde{x}$
            update the partitions
            update the distance matrix
    End the for loop
```

Figure 1: Pseudo-code of the agglomerative information bottleneck algorithm[1]- $d_{i,j}$ is the distance between partition *i* and partition *j*

called *P*, which in our data set is the product price. After finding the minimum value sets, we look into set *P* as well and choose the best minimum value based on both minimum cost values and the extra information set, i.e the two products that have the minimum price difference. The pseudo-code of the modified AIB method is in figure 2.

```
Input : joint probability p(x,y)
output : partition of X into m cluster
Inititialize:
    $\tilde{X} = X$
    $d_{i,j} = (p(\tilde{x_i}) + p(\tilde{x_j}))D_{JS}[p(y|\tilde{x_i}), p(y|\tilde{x_j})]$
Loop:
    For all the partitions in X
            find the k smallest $d_{i,j}$
            choose the best $d_{i,j}$ based on the extra information set P
            merge $\tilde{x_i}, \tilde{x_j}$ into $\star\tilde{x}$
            update the partitions
            update the distance matrix
    End the for loop
```

Figure 2: Pseudo-code of the modified agglomerative information bottleneck algorithm. In our experiment, set *P* is the product prices. After finding *k* minimum values, we check for the minimum price difference between the clusters and make a merging decision based on both the minimum distance and the minimum price difference.

3 Experiment

In this section we present our findings on applying agglomerative information bottleneck method and the modified AIB.

3.1 The Evaluation Method

Evaluating clustering performance is a very hard task; there are different approaches. One approach is to use labeled data to find the performance of the clusterer. In classification, labeled data is used to evaluate the performance of a classifier. The use of labeled data to measure the performance of the clusterer is not the best approach. Classifiers and clusterers focused on different features of the data.

In our approach we use labeled data but not the same as classifiers. After applying the clustering method on the data set, the output of the clusterer is not necessarily the same as the original clusters. We need to relabel the new clusters to find the corresponding cluster in the original set. When the re-labeling is done, the clustering error is evaluated based on the difference of the original clusters and the re-labeled clusters. As an example if the following sets are the original clusters and the new clusters after clustering:
$Originalclusters = \{1, 1, 1, 1, 2, 2, 2, 3, 3\}$
$Newclusters = \{2, 2, 2, 2, 3, 3, 3, 1, 1\}$
Then, the clustering error in this example is zero. The only change in this example is in name of the culsters. The way we do the relabeling , is to use the labeled data and new clusters and find the members of each new cluster based on the original clusters and then re-labeled the new clusters based on the average of it's members.

3.2 Datasets

We used the product titles from Amazon, Best buy, Target, Future shop as our labeled data set. We have two sets of data; one including unique 77 and 66 electronic devices and there are 3 or 4 variations of the same product from different sites. As an example, there are three variations of 55-inch Bravia Sony TV in our data list. Our goal is to cluster the same products together. Therefore, after applying the clustering, we have to have the three 55-inch Bravia Sony TVs in the same cluster.

4 Experiment Results

We run the both AIB and modified AIB on our data set and the resulting performance was not promising. In the first run that we used AIB method, the clustering error was 48%. In the second run, we used the modified AIB method and the clustering error decreased to 38% which is a huge improvement but not good enough for a clusterer. We used the Naive Bayesian classifier to measure the performance of AIB and modified AIB against another approach. Bayesian classifiers are statistical classifiers. They can predict the membership probability, i.e. the probability of an instance belongs to a certain class. Bayesian classifiers work based on Bayes rule. The naive term comes from the fact that, in Naive Bayesian classifiers the assumption is that features of a class are independent from each other, and there is no dependency among the features. Although this assumption is not a very smart one but Naive Bayesian outperform many sophisticated classifiers. In our experiment, the Naive Bayesian classifier accuracy was 90.91%, i.e. the amount of error was only 10%.

5 Discussion and Future Works

According the the results of our experiment, in dealing with data sets with not sufficinet text contents, agglomerative information bottleneck method does not perform very well. Adding extra information in merging step improves the performance of agglomerative information bottleneck method by 10%. To enhance the performance, one can add more text to the content of each data point. As a part of future investigation on the present data set, one can add the product description to the data set and apply both AIB and modified AIB. A better performance is predicted comparing to the present results.

The reason behind the bad performance of Agglomerative information bottleneck method is the fact that AIB summarizes the text content of a data set in order to find the important features of each data point. The problem arises when dealing with data sets such as product titles which contains on average 10 words. Finding important features of a short document in order to compress is an erroneous task; therefore, AIB or modified AIB are not the right choice for clustering products based on their titles. On the other hand, Naive Bayesian Classifier seems to perform decently on the present data set. As part of the future work, assigning more weight to some of the features in this data set can be a good candidate to enhance the performance of the Naive Bayesian classifier.

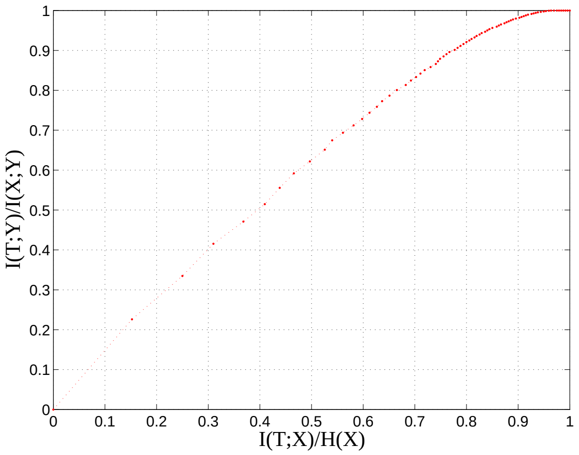

Figure 3: In this figure, the results of modified agglomerative information bottleneck method is shown. The relative mutual information versus the normalized mutual information between the partitions and the original data is presented in the figure. In AIB, we are interested to cluster the data set in a way that each cluster has the maximum amount of mutual information among its member. AIB is interested in finding partitions so that the difference between the mutual information of the partitioned data and non-partition data is minimum or the relative mutual information is near one. This figure is a representative of quality of quantization of the original variable.

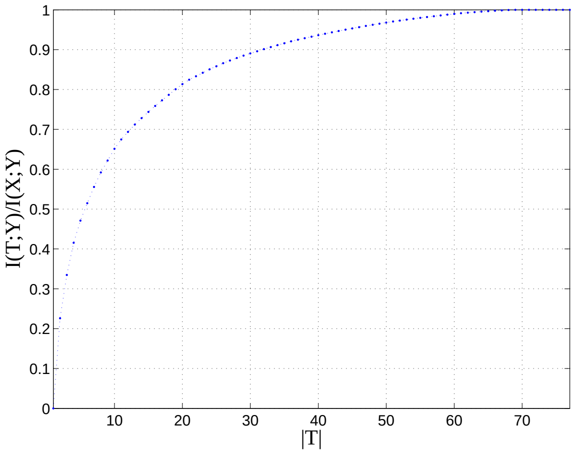

Figure 4: This figure shows the relative mutual information versus the number of partitions. As you can see, the perfect partition number is more than 50, when the relative mutual information is almost one. In out data set, the true partition number is 20, but due to the fact that AIB is not the best fit for the presented data set, AIB proposes more than 50 partitions.

References

[1] Tishby, N. & Pereira, F.C. & Bialek W. (1999) The Information Bottleneck Method. *Proc. of 37-th Allerton Conference on Communication and Computation*.

[2] Tishby, N. & Pereira, F.C. & Bialek W. (1998) The Information Bottleneck Method: Extracting Relevant Information from Concurrent Data *NEC Research Institute TR.*